LETTER

PDAA3C: An A3C-Based Multi-Path Data Scheduling Algorithm*

Teng LIANG[†], **Ao ZHAN**[†], **Chengyu WU**[†a)], ***Nonmembers***, ***and*** **Zhengqiang WANG**[††], ***Member***

SUMMARY In this letter, a path dynamics assessment asynchronous advantage actor-critic scheduling algorithm (PDAA3C) is proposed to solve the MPTCP scheduling problem by using deep reinforcement learning Actor-Critic framework. The algorithm picks out the optimal transmitting path faster by multi-core asynchronous updating and also guarantee the network fairness. Compared with the existing algorithms, the proposed algorithm achieves 8.6% throughput gain over RLDS algorithm, and approaches the theoretic upper bound in the NS3 simulation.

key words: *MPTCP, data scheduler, throughput, network fairness*

1. Introduction

Multipath transmission control protocol (MPTCP) [1] is an extension of the traditional TCP, which enables full use of the device's multiple interfaces [2] and increases transmission efficiency, link fairness, and throughput. Currently, users' demands are increasing quickly for high network bandwidth and low end-to-end transmission delay, because of the rapid development of novel technologies such as virtual reality and real-time live broadcast [3]. Mobile devices now own multiple network interfaces to different network access technologies, such as WiFi and cellular. MPTCP has thus got much attention due to it aggregates multiple subflows capacity and maintains single-path failures.

Scheduling is a core component of MPTCP, which controls the amount of traffic transmitted over distribution package and maintains link fairness. To improve the subflow throughput [4] and fairness of MPTCP, several scheduling algorithms have proposed (e.g., Average-RTT and Fastest-RTT [5]) based on traditional MPTCP scheduling algorithm (for example, Round-robin [6]) distribute packets by polling when packets come from application layer. Heterogeneous multi-subflow network (HMN) is different in sub-flow service indicators vary greatly and subflow available bandwidth, which achieved desired performance hardly for MPTCP. On the one hand, MPTCP faces difficulties in distributing packets to heterogeneous multiple paths reasonably. Sub-flow with smaller bandwidth may severely degrade the performance of other sub-flows in a MPTCP connection. On the other hand, it is hard for MPTCP to judge the subflow status accurately in HMN. Therefore, it is an important topic for researchers to formulate a reasonable scheduling strategy so that users feel higher bandwidth, lower end-to-end delay and maximize throughput in HMN.

Recently, SB-FPS [7], a novel scheduling algorithm which schedules the bottleneck scenarios shared data, improves 6% throughput over the default MPTCP by varing the window size of each sub-flow and proves that it is properly to distribute subflow data according to the state of subflow. The combination of computer network optimization research and deep reinforcement learning (DRL) [8] in the era of artificial intelligence, which is optimized for MPTCP.

Some scheduling algorithms based on DRL like RLDS [9], Peekaboo [10], and GAPS [11] aims to improve the MPTCP throughput performance in a heterogeneous network. This is an intuitive deduction that MPTCP benefits from DRL because DRL captures and analyzes status of MPTCP subflow precisely. Applying DRL to increase MPTCP performance, there are two issues that need to be thought. First of all, scheduling should be redesigned for MPTCP to take advantage of DRL in HMN. That means when choosing a DRL algorithm, it needs to consider the MPTCP scheduling problem is a continuous problem. Meanwhile, to assign number of packets accurately, the evaluation of sub-flow status needed to redesign.

In this letter, we propose a multipath scheduling algorithm for MPTCP by asynchronous advantage Actor-Critic(A3C) and a new criterion for the evaluation of subflow, named MPTCP-PDAA3C. The proposed algorithm can achieve 26.6%-106% throughput gain over existing algorithms, and improves fairness among sub-flows in NS3.

2. PDAA3C-MPTCP Design

2.1 DRL Framework

The MPTCP scheduling process can be modeled as a discrete optimization problem, and A3C can obtain better convergence properties than other RL algorithms for the discrete optimization problem [12]. However, it is diffcult for MPTCP to combine with the A3C algorithm that requires redesigning the STATE, ACTION and REWARD modules.

Manuscript received June 18, 2022.
Manuscript revised August 25, 2022.
Manuscript publicized September 13, 2022.

[†]The authors are with School of Information Science and Engineering, Zhejiang Sci-Tech University, Hangzhou, 310018 P. R. China.

[††]The author is with School of Communication and Information Enginnering, Chongqing University of Posts and Telecommunication, Chongqing, 400065 P. R. China.

*This work was supported by the Fundamental Research Funds of Zhejiang Sci-Tech University under grant 2021Q029 and Key Laboratory of Universal Wireless Communications (BUPT) under grant No. KFKT-2018101, Ministry of Education, P. R. China.

a) E-mail: jerry916@zstu.edu.cn (Corresponding author)

DOI: 10.1587/transinf.2022EDL8052

Therefore, PDAA3C is proposed to achieve the redesign of three modules. The PDAA3C puts Actor-Critic into multiple threads for synchronous training, effectively utilizing computer CPU resources and improving training effectiveness. Multi-agent in PDAA3C interact with the environment E over multiple discrete time steps within a DRL framework. Actor takes the state S_t as input and outputs the corresponding action A_t by $\pi(a_t|s_t;\theta)$, and Critic takes the state-action (s_t, a_t) as input and outputs the corresponding $V(s_t, a_t; \theta_v)$. In return, the agent receives the next state S_{t+1} and receives a scalar reward $V(s_t, a_t) = \mathrm{E}[R_t|s_t, a]$, where R_t is the value of choosing an action a in state s. $\pi(a_t|s_t;\theta)$ is the retention strategy, $V(s_t;\theta_v)$ is the estimated value function, and the update gradient of strategy π is formulated as:

$$\nabla'_\theta \log \pi(a_t|s_t;\theta')A(s_t, a_t;\theta,\theta_v) \tag{1}$$

where the dominant function $A(s_t, a_t;\theta,\theta_v)$:

$$\sum_{i=0}^{k-1}\gamma^i r_{t+i} + \gamma^k V(s_{t+k};\theta_v) - V(s_t;\theta_v) = R_t - V(s_t;\theta) \tag{2}$$

where R_t is the return of all states, k varies with state and the maximum value is t_{max}, θ' is the parameter of the policy π and θ'_v is the parameter of the state value function.

Critic uses $J(\theta)$ to estimate the state, which is the optimization of the state function V.

$$J(\theta) = \frac{1}{N}\sum_{1}^{N}(\sum_{i=0}^{k-1}\gamma^i r_{t+i} + \gamma^{T_n} V(s_{t+k};\theta_v) - V(s_i;\theta))^2 \tag{3}$$

where N is batch size, T_n are steps, $V(s_i;\theta)$ is actual state value.

Next, we design the states, actions and rewards of PDAA3C-MPTCP:

STATE: At epoch t, $S_t^{i,k} = [S_t^{1,1}, S_t^{i,k}, \cdots, S_t^{N,k}]$, St = $[ST_t^i, TT_t^i, C_t^i, TP_t^i, PLR_t^i]$, where N is the fluctuation Total number of the fluctuant subflows, k is the link state number of the fluctuant subflows i, and $S_t^{i,k}$ is the substream i in link state k at time t. ST_t^i, TT_t^i, C_t^i, TP_t^i, PLR_t^i are the data confirmed from subflow i, such as free bandwidth, throughput, congestion window, number of packets on the current link, and packet loss rate. In experimentations, the number of packets on the current subflow is calculated by the throughput and Packet-size (P), the throughput is the amount of data sent by the link at time t.

The key parameters may have a significant impact on the end-to-end performance [13], which are selected into the state of PDAA3C and considered in the design of some related projects. To improve the accuracy of judging the quality of subflows, the data of all subflows are considered in this work, when designing the state space. Noting that the values of these parameters are all measured in the past epoch $t-1$.

ACTION: Action at time period t, $a_t = [A_{t^1}, \cdots, A_{t^k}]$, where A_{t^k} is to select a link to allocate packets at each epoch t. The positive, negative and zero actions result in the selection of the optimal subflow path, the regular sub-flow path and the very poor subflow. Respectively, PDAA3C takes action on an optimal MPTCP subflow of N MPTCP subflows at each epoch t.

REWARD: The reward at epoch t, $r_t = \sum_i^N U(i,t)$, where $U(I,t)$ gives the reward for MPTCP the fluctuant subflow i. Caculating rewards have many different functions (e.g. throughput, latency, alpha fairness). The reward should be designed according to the actual needs of the upper-layer application. In implementations, we choose the widely used reward function $U(I,t) = \log TT_t^i$ for the problem of accurately reflecting the transmission quality [13].

2.2 Subflow Quality Evaluation

The subflow quality evaluation criteria in step 2 are combined with ST_t^i, TT_t^i, C_t^i, TP_t^i, PLR_t^i, and RTT_t^i, calculating transmission quality Q of the MPTCP path.

$$Q_t^i = \frac{\gamma\alpha(1-PLR_t^i)}{ST_t^i} + \frac{\eta C_t^i}{RTT_t^i - TT_t^i} \tag{4}$$

$$ST_t^i = \frac{C_t^i}{RTT_t^i - TT_t^i} \tag{5}$$

where $\gamma = 0.7$, $\eta = 0.3$, α is the packet size, $1 - PLR_t^i ST_t^i$ is the time required for the current subflow i to transmit a packet of size P. This subflow evaluation criterion most truly show the current transmission quality of the subflow after reviewing data. PDAA3C-MPTCP combines with the subflow quality evaluation criteria and selects the optimal subflow to transmit packets at each time t, to improve the subflow throughput obviously after DRL training.

2.3 PDAA3C Structure and Algorithm

Figure 1 shows the structure of PDAA3C-MPTCP algorithm, which includes an MPTCP Server, an MPTCP Client, N Sub-flows and N Sub-flows states. In order to select the optimal subflow for packet transmission for each decision, PDAA3C-MPTCP uses actor network to select subflows for data transmission and critic network to score Actor Network's behavior. In MPTCP client, PDAA3C-MPTCP-Scheduler is consistent with other MPTCP scheduling algorithms, realizing data scheduling and distribution with MPTCP client.

The design goal of PDAA3C-MPTCP-Scheduler is to

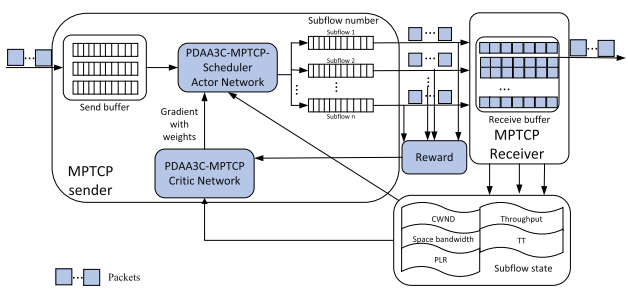

Fig. 1 System framework diagram of PDAA3C-MPTCP algorithm

Algorithm 1 Path Dynamics Assessment Asynchronous Advantage Actor-Critic scheduling algorithm

Input: θ, θ_v, $d\theta$, $d\theta_v$, t_{start}, R, PLR_t^i, Q_t^i, ST_t^i, RTT_t^i, TT_t^i, C_t^i
Output: θ', θ'_v, Q_{t+1}^i, ST_{t+1}^i, a_{t+1}, s_{t+1}, $d\theta'$, $d\theta'_v$
Initialization: $d\theta \leftarrow 0$, $d\theta_v \leftarrow 0$, γ=0.7, α=1000, η=0.3, t_{start}=0, $R = 0$
repeat
Synchronize thread-specific parameters $\theta' = \theta$, $\theta'_v = \theta_v$
Get State S_t, $workers_{net}$Upload gradient reset and analyze link statusQ_t^i
$Q_t^i = \frac{\gamma\alpha(1-PLR_t^i)}{ST_t^i} + \eta ST_t^i$
$ST_t^i = \frac{C_t^i}{RTT_t^i - TT_t^i}$
transmit the packet by Q_t^i best subflow i
repeat
Perform a_t according to policy $\pi(a_t|s_t;\theta')$
Receive reward r_t and new state s_{t+1}
$t \leftarrow t+1$, $T \leftarrow T+1$
until terminal s_t or $t - t_{start} == t_{max}$

$$R = \begin{cases} 0 & for \quad terminal \quad s_t \\ V(s_t, \theta'_v) & for \quad non-terminal \quad s_t \end{cases}$$

for $i \in t-1, \cdots, t_{start}$ **do**
  $R \leftarrow r_i + \gamma R$
  Accumulate gradients $\theta' : d\theta' \leftarrow d\theta + \nabla'_\theta \log \pi(a_t|s_t;\theta')(R - V(s_i;\theta'_v))$
  Accumulate gradients $\theta'_v : d\theta'_v \leftarrow d\theta_v + \partial(R - V(s_i;\theta'_v))^2/\partial\theta'_v$
end for
Perform asynochronous update of θ using $d\theta$ and of θ_v using $d\theta_v$
until $T > T_{max}$

select the optimal transmission path among multiple subflows. The detailed description of its multi-path management and scheduling are as follows:

i) Packet distribution: MPTCP server concurrently distributes data packets to established sub-flows, transmits them to MPTCP client, and obtains the current state $S_t^{i,k}$ of each sub-flow.

ii) Subflow quality ranking: PDAA3C-MPTCP-Scheduler combines with the subflow quality evaluation criteria and $S_t^{i,k}$ to sort the quality of all sub-flows.

iii) Optimal scheduling: PDAA3C-MPTCP-Scheduler distributes data packets to the optimal sub-flow, and obtains the current state $S_t^{i,k+1}$ of each sub-flow.

iiii) Special environment: It is assumed that sub-flow congestion does not occur in this path scheduler, which causing the sub-flow without transmitting data.

Sum up, the proposed PDAA3C algorithm is shown as Algorithm 1.

2.4 PDAA3C Feasibility

We perform a lot of tests for the execution steps and decision time of the PDAA3C and RLDS algorithm on a PC with i7-10875H CPU and 32GB RAM. We insert timestamps in PyCharm 2020.3.5 x64 and count the execution time of DRL algorithms for 100, 200, 300, and all the way to 2000 steps.

Figure 2 shows that the computational cost of the PDAA3C algorithm is 18.8ms/times-40.7ms/times, and the computational cost of the RLDS algorithm is 20.4ms/times-63.1ms/times in 0-2000 execution steps. We see that PDAA3C has better processing delay stability than RLDS,

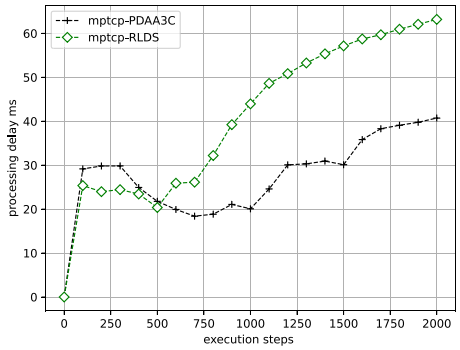

Fig. 2 The computational cost of PDAA3C and RLDS algorithm

because PDAA3C gets better convergence properties than RLDS for the discrete problems of MPTCP. The two subflow processing delays are 30ms and 60ms in the 70s simulation experiment of RLDS, and the computational cost of the RLDS is 20.4ms/times-43.9ms/times. Our simulation considers capturing the throughput every 100ms, i.e., 1 subflow selection at the sender and receiver in every 100ms, and the interval of sub-flow throughput capture is freely setting on NS3. The ACK acknowledgement is synchronized when sub-flow selection is performing, 100ms $\gg$ 29.8ms, 29.8ms is the processing delay for each of the 200 execution steps, and 100ms satisfies the maximum decision delay of the PDAA3C algorithm. In summary the processing delay of PDAA3C satisfies the requirement of acquiring ACK every round trip.

3. Performance Evaluation

The performance evaluation of PDAA3C-MPTCP has been determined on the network simulator NS 3.29 and PyCharm 2020.3.5 x64. The basic MPTCP module, Round-Robin and Fastest-RTT scheduling algorithms are implemented in the network simulator NS 3.29, PDAA3C-MPTCP and mptcp-RLDS is implemented in PyCharm. In addition, the congestion control algorithm is MPTCP-BBR, which is used to prevent unreliable experimental data because of serious congestion on the path during the experiment. We use Jain's fairness index to compare the link fairness. The equation of Jain's fairness index:

$$FI = (\sum_{i=1}^{n}(T_i/O_i))^2/(n\sum_{i=1}^{n}(T_i/O_i)^2) \tag{6}$$

where FI is the fairness index, T_i is the transmission capacity of the i subflow in the network, and O_i is the actual throughput of the i subflow when all n subflows are working. The value of FI is in the range $[1/n, 1]$. When $FI = 1$, the system is absolutely fair. When $FI = 1/n$, the system is completely unfair.

The subflow throughput simulation results are shown in Fig. 3, comparing mptcp-RLDS, mptcp-round-robin, mptcp-fastest-rtt, and mptcp-pdaa3. In Fig. 3, the subflow throughput of the mptcp-pdaa3c algorithm is better than

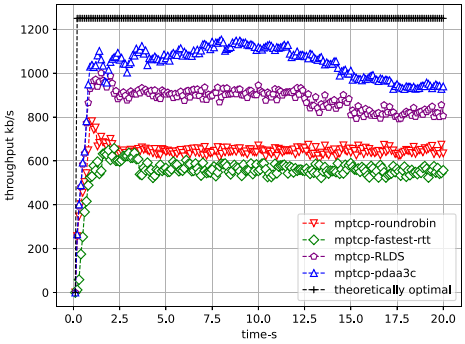

Fig. 3 Comparison of subflow throughput results of the four algorithms

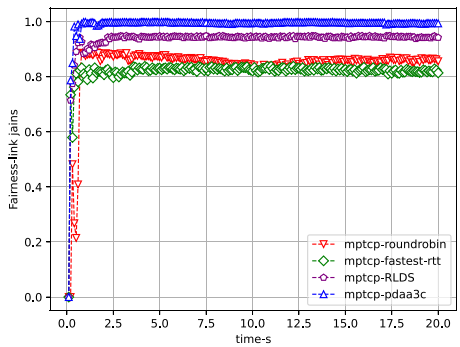

Fig. 4 Subflow fairness simulation results of the four algorithms

mptcp-RLDS 8.6%, mptcp-round-robin 49.6% and mptcp-fastest-rtt 52.6%, and approaches to the theoretically optimal value of throughput. Mptcp-pdaa3c combines the subflow quality evaluation criteria and A3C of the DRL to select the optimal subflow, when the data packet needs to be transmitted in the multipath transmission simulation experiment. Therefore, mptcp-pdaa3c is able to obtain higher throughput.

The subflow fairness comparison results of mptcp-RLDS, mptcp-round-robin, mptcp-fastest-rtt and mptcp-pdaa3c are shown in Fig. 4. The subflow fairness of mptcp-pdaa3c is always better than others and approaches 1. The characteristic of mptcp-pdaa3c is that when the packet needs to be transmitted, the optimal subflow is selected by the proposed scheduling algorithm and the subflow quality evaluation criteria, so mptcp-pdaa3c subflow fairness is the best among the four algorithms.

4. Conslusion

In this letter, we propose a multi-path scheduling algorithm PDAA3C based on a DRL to enhance the throughput of multi-path transmission and ensure the fairness of each link. As far as we know, it is the first time to combine the A3C algorithm of DRL with the MPTCP protocol and carry out simulation implementation.

PDAA3C-MPTCP includes two additional modules, namely mptcp-pdaa3c-scheduler-Actor Network and mptcp-pdaa3c-Critic Network. Mptcp-pdaa3c-scheduler-Actor Network combines the link performance status to distribute packets concurrently to the optimal sub-flow path in the sub-flow link. Mptcp-pdaa3c-Critic Network judges and scores the decisions of mptcp-pdaa3c-scheduler-Actor Network, and then prompts it to choose the optimal sub-flow path each time. The simulation results show that the transmission performance of mptcp-pdaa3c is better than existing algorithms, and the link fairness of mptcp-pdaa3c is better than that of mptcp-fastest-rtt, mptcp-roundrobin in the case of asymmetric paths.

References

[1] M. Handley, O. Bonaventure, C. Raiciu, and A. Ford, "TCP extensions for multipath operation with multiple addresses," IETF RFC 6824, 2013.
[2] L. Ming, A. Lukyanenko, Z. Ou, A. Ylä-Jääski, S. Tarkoma, M. Coudron, and S. Secci, "Multipath transmission for the internet: A survey," IEEE Communications Surveys & Tutorials, vol.18, no.4, pp.2887–2925, 2016.
[3] H. Zhang, Z. Yang, and P. Mohapatra, "Wireless access to ultimate virtual reality 360-degree video: poster abstract," IoTDI '19: International Conference on Internet-of-Things Design and Implementation, pp.271–272, 2019.
[4] H. Shi, Y. Cui, X. Wang, Y. Hu, M. Dai, F. Wang, and K. Zheng, "STMS: Improving MPTCP throughput under heterogeneous networks," Proc. 2018 USENIX Conference on Usenix Annual Technical Conference, Boston, MA, USA, p.719–730, 2018.
[5] R. Lübben and J. Morgenroth, "An odd couple: Loss-based congestion control and minimum RTT scheduling in MPTCP," IEEE 44th Conference on Local Computer Networks, pp.300–307, 2019.
[6] C. Paasch, S. Ferlin, O. Alay, and O. Bonaventure, "Experimental evaluation of multipath TCP schedulers," Proc. 2014 ACM SIGCOMM Workshop on Capacity Sharing Workshop, New York, NY, USA, p.27–32, 2014.
[7] W. Wei, K. Xue, J. Han, Y. Xing, D.S.L. Wei, and P. Hong, "BBR-based congestion control and packet scheduling for bottleneck fairness considered multipath TCP in heterogeneous wireless networks," IEEE Trans. Veh. Technol., vol.70, no.1, pp.914–927, 2021.
[8] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," 2017 IEEE International Conference on Robotics and Automation (ICRA), pp.3389–3396, 2017.
[9] J. Luo, X. Su, and B. Liu, "A reinforcement learning approach for multipath TCP data scheduling," IEEE 9th Annual Computing and Communication Workshop and Conference, pp.0276–0280, 2019.
[10] H. Wu, O. Alay, A. Brunstrom, S. Ferlin, and G. Caso, "Peekaboo: Learning-based multipath scheduling for dynamic heterogeneous environments," IEEE J. Sel. Areas Commun., vol.38, no.10, pp.2295–2310, 2020.
[11] B. Liao, G. Zhang, Z. Diao, and G. Xie, "Precise and adaptable: Leveraging deep reinforcement learning for GAP-based multipath scheduler," IEEE IFIP Networking Conference, Paris, France, pp.154–162, 2020.
[12] V. Mnih, A.P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," International Conference on Machine Learning, pp.1928–1937, PMLR, 2016.
[13] K. Winstein and H. Balakrishnan, "TCP ex machina: Computer-generated congestion control," Proc. ACM SIGCOMM, New York, NY, USA, vol.43, no.4, pp.123–134, 2013.